Mobile Application Development

Higher Diploma in Science in Computer Science

Produced by

Eamonn de Leastar (edeleastar@wit.ie)

Department of Computing, Maths & Physics
Waterford Institute of Technology
http://www.wit.ie
http://elearning.wit.ie

Waterford Institute of Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit

Modeling & Testing witpressapp

Blog

- Each user can have a blog

- A blog consists of multiple 'Posts'

- Each Post consists of:

    - Title

    - Content

User

```java
@Entity
public class User extends Model
{
  public String firstName;
  public String lastName;
  public String email;
  public String password;

  @OneToMany(cascade=CascadeType.ALL)
  public List<Post> posts;

  public User(String firstName, String lastName,String email, String password)
  {
    this.firstName = firstName;
    this.lastName = lastName;
    this.email = email;
    this.password = password;
    posts = new ArrayList<Post>();
  }

  public void addPost (Post post)
  {
    posts.add(post);
  }

  public static User findByEmail(String email)
  {
    return find("email", email).first();
  }

  public boolean checkPassword(String password)
  {
    return this.password.equals(password);
  }
}
```

Post

```java
@Entity
public class Post extends Model
{
  public String title;
  @Lob
  public String content;

  @OneToMany(cascade = CascadeType.ALL)
  public List<Comment> comments;

  public Post(String title, String content)
  {
    this.title = title;
    this.content = content;
    this.comments = new ArrayList<Comment>();
  }

  public void addComment(Comment comment)
  {
    comments.add(comment);
  }

  public String toString()
  {
    return title;
  }
}
```
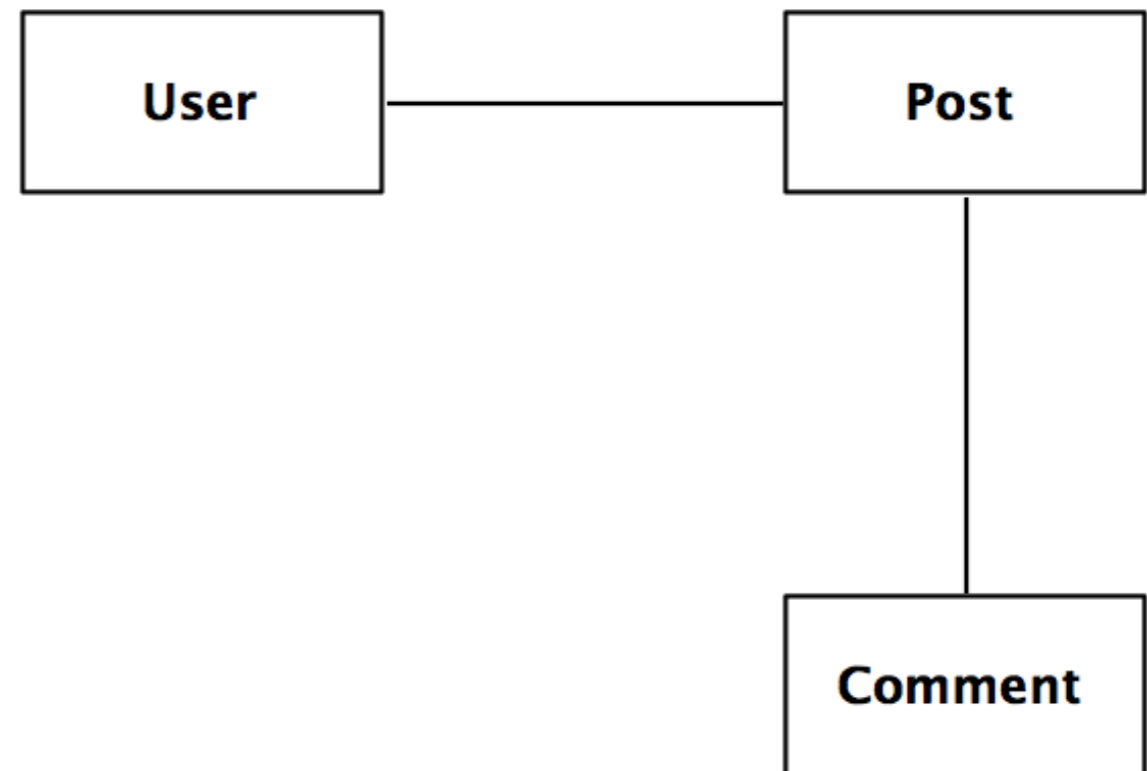
Comment

```java
@Entity
public class Comment extends Model
{
  public String content;

  public Comment(String content)
  {
    this.content = content;
  }
}
```
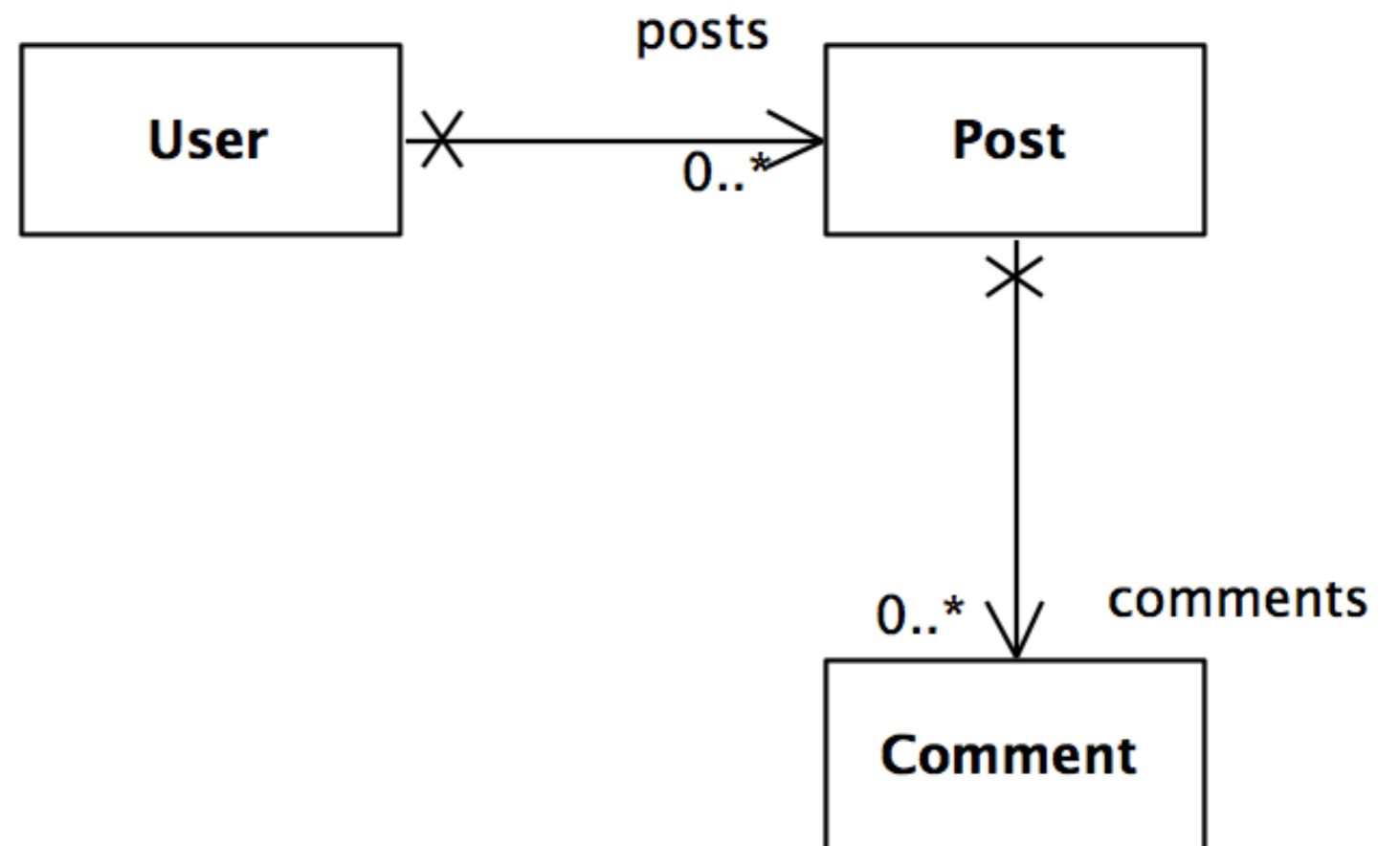
Model

- Each class represented by a simple rectangle

- If a class is 'aware' of another class, then draw a line connecting them.

- These lines are called 'associations'

Model

- Enrich the associations with extra information

  - Navigability

  - Role name

  - Cardinality
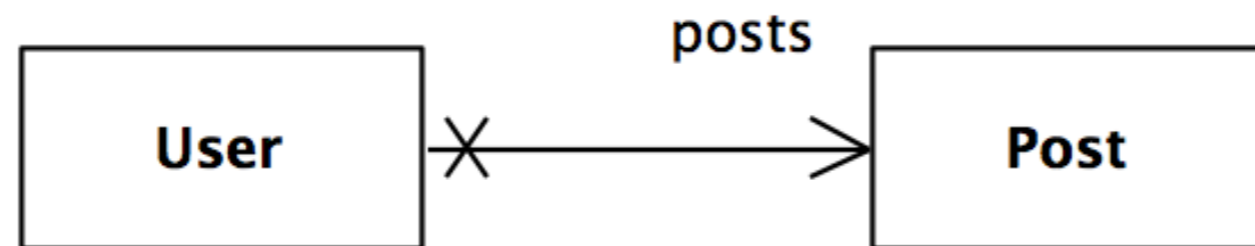

posts

User ⎯⨯⎯→ Post
0..*

Post ⨯
│
0..* ↓ comments
Comment

Navigability

- Indicates whether one class can be 'reached' from another

- eg:

  - User maintains a collection of posts

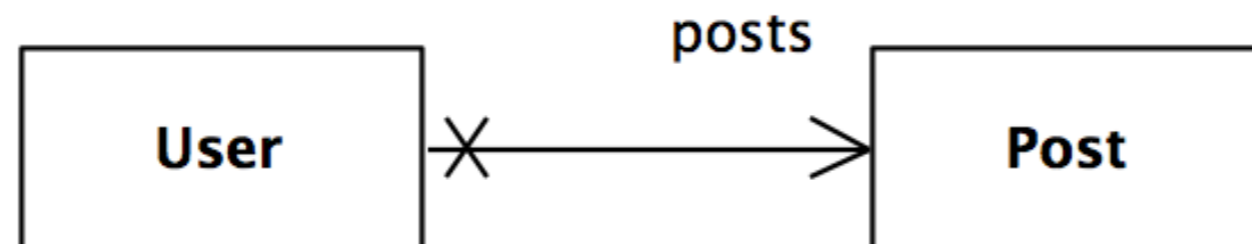 - However, posts (in this model at least) are unaware of the user they belong to
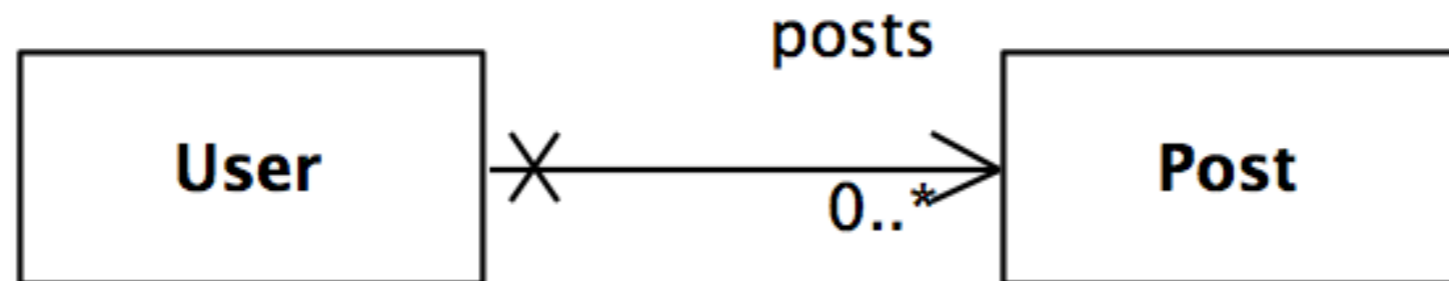


```
public class User extends Model
{
  //...
  @OneToMany(cascade=CascadeType.ALL)
  public List<Post> posts;
  //...
}
```

```
public class Post extends Model
{
  //..
}
```

Role Name

- The name of the attribute used to traverse the navigable path

- eg:

  - User Post collection is called 'posts'

  - Drawn at the 'other' end of the association


```java
public class User extends Model
{
  //...
  @OneToMany(cascade=CascadeType.ALL)
  public List<Post> posts;
  //...
}
```

```java
public class Post extends Model
{
  //..
}
```

10

Cardinality

- How objects will be managed during the application lifecycle

  - 1 - a one-to-one relationship

  - 0..* - a zero-to-many relationship

- OneToMany == 0..* in modeling terms (as the collection can be empty)
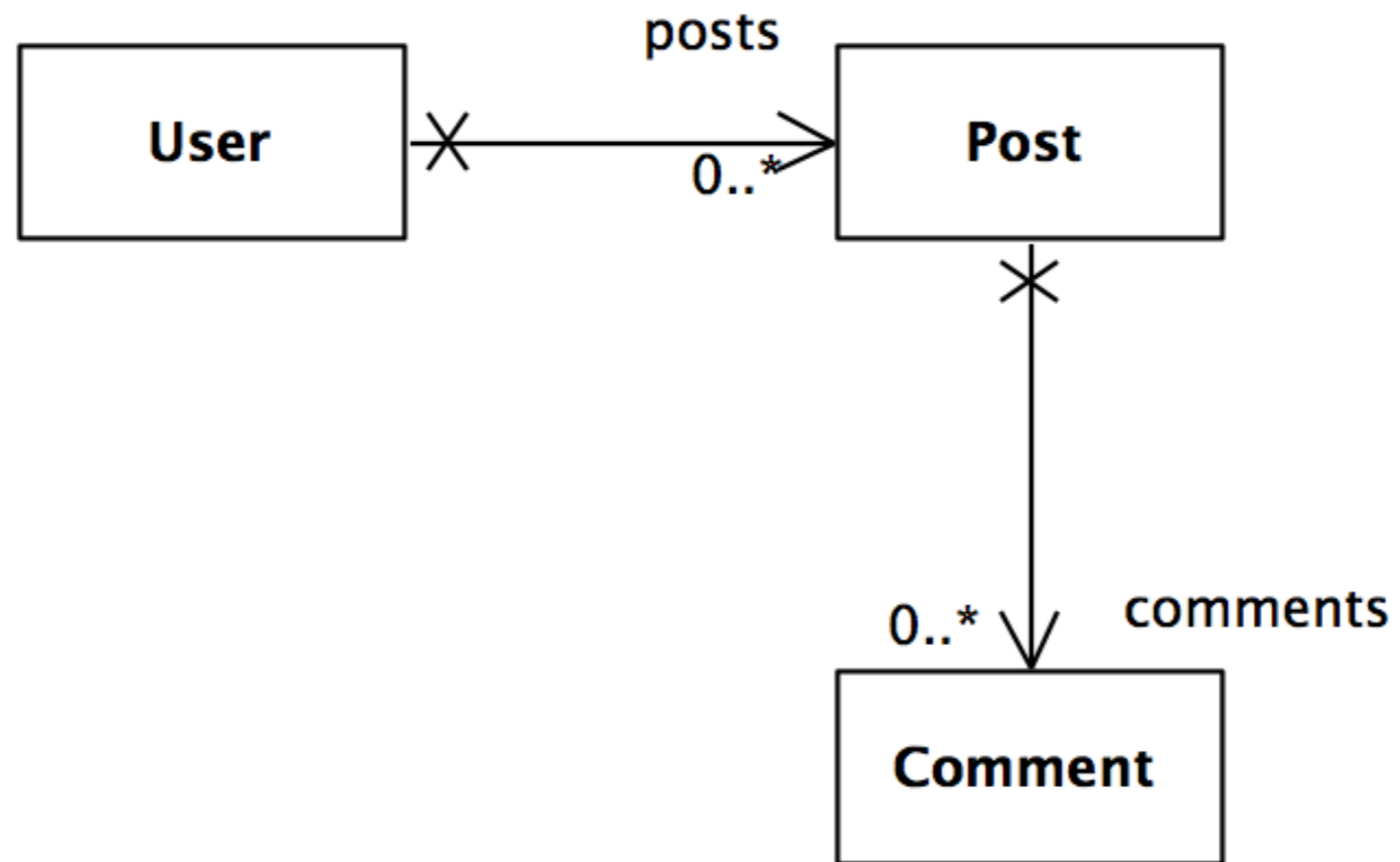


```
public class User extends Model
{
  //...
  @OneToMany(cascade=CascadeType.ALL)
  public List<Post> posts;
  //...
}
```

```
public class Post extends Model
{
  //..
}
```

Witpress Model

Testing witpress (1)

- Exercise 1

  - Write these tests

Testing witpress (3)

- **Exercise 2**

  - Currently a Blog is not modeled - a blog just a collection of posts. Model a Blog as a first class object, developing tests to verify the implementation as you go. Do this in two versions:

    - Each user can have a single blog.

    - Each user may have zero or more blogs

  - Dont worry about the UI, just focus on the tests

Waterford Institute *of* Technology
INSTITIÚID TEICNEOLAÍOCHTA PHORT LÁIRGE

eLearning
support unit